Haozhu Wang

Project 2.1 Report

Numeric Values:

| | outlook |
|---|---|
| sunny | 5 |
| overcast | 4 |
| rainy | 5 |

| | Humidy |
|---|---|
| Min | 65 |
| Max | 96 |
| Mean | 81.643 |
| StdDev | 10.285 |

| | Windy |
|---|---|
| TRUE | 6 |
| FALSE | 8 |

| | Play |
|---|---|
| yes | 9 |
| no | 5 |

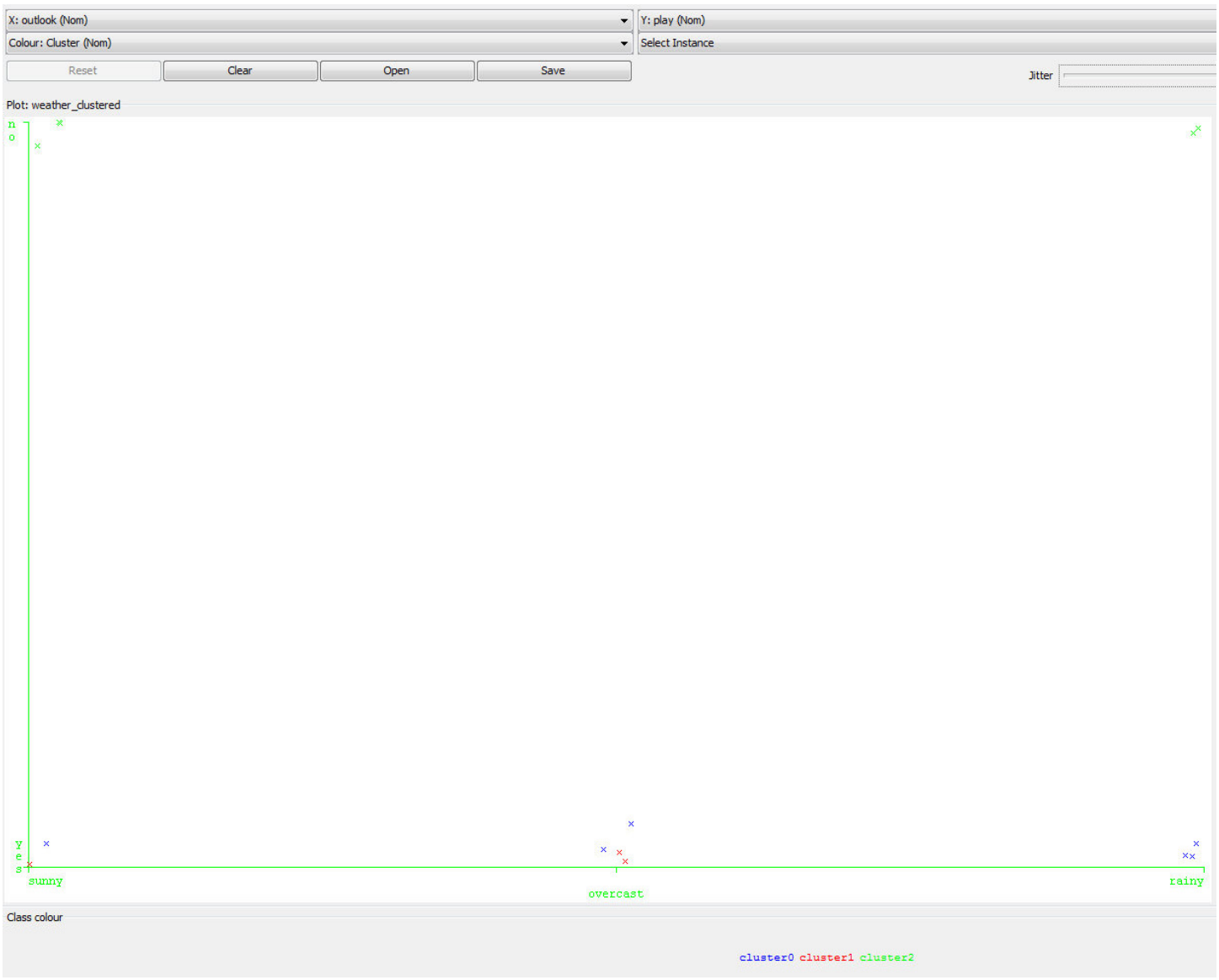
| | Temperature |
|---|---|
| Min | 64 |
| Max | 85 |
| Mean | 73.571 |
| StdDev | 6.572 |

Clustering:

The first clustering algorithm I ran is just simple KMeans in the non-normalized aarf file dataset.

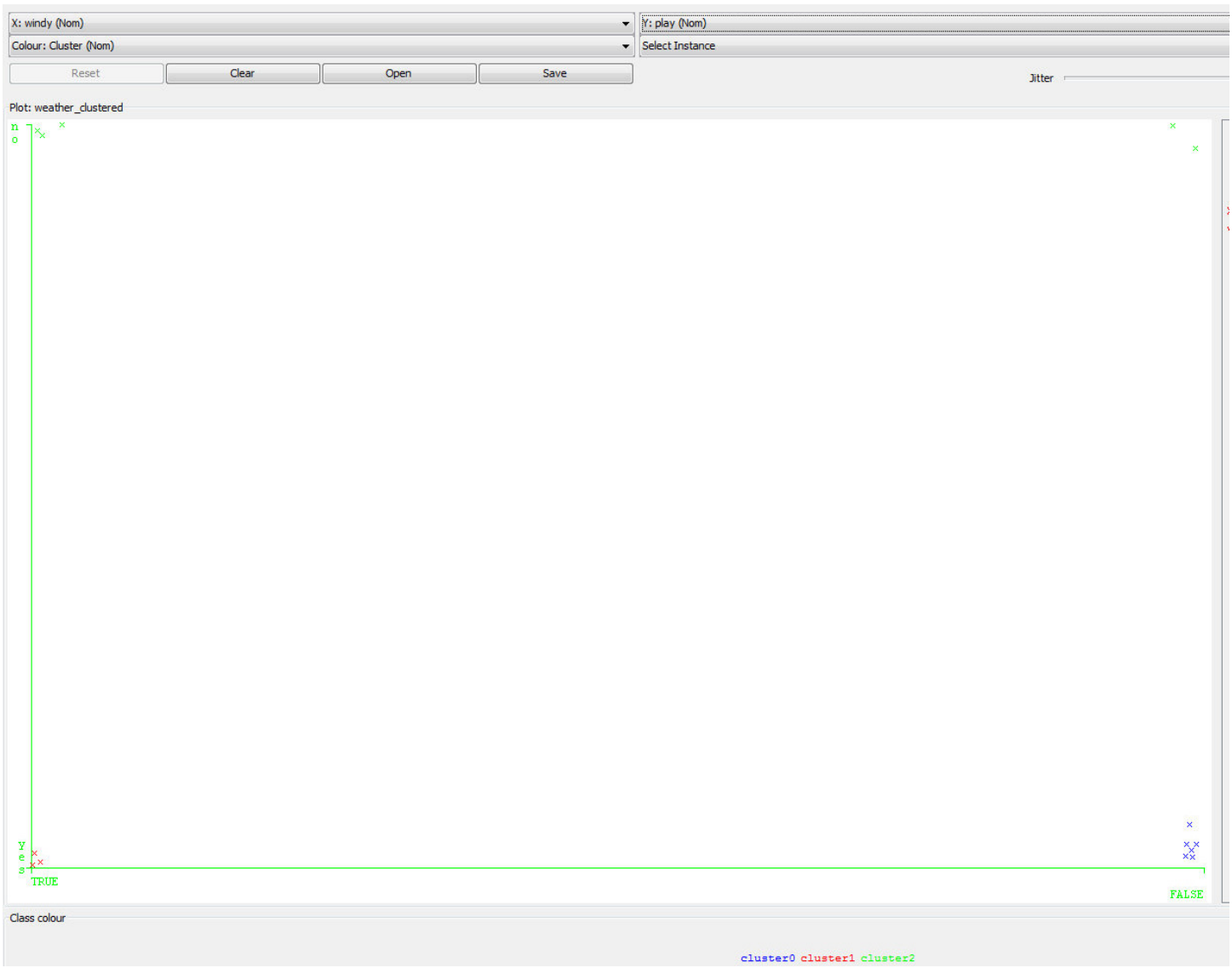
```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    evaluate on training data

=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 10.36566100599102
Missing values globally replaced with mean/mode

Cluster centroids:
                        Cluster#
Attribute       Full Data          0          1          2
                     (14)         (6)        (3)        (5)
============================================================
outlook             sunny      rainy   overcast      sunny
temperature       73.5714    74.3333    70.3333       74.6
humidity          81.6429    81.1667         75       86.2
windy               FALSE      FALSE       TRUE       TRUE
play                  yes        yes        yes         no


Clustered Instances

0       6 ( 43%)
1       3 ( 21%)
2       5 ( 36%)
```

One of the first interesting graphs I encountered is *outlook vs. play*:




For this graph I turned on the Jitter so the data points can be separated. I notice here that for “no” in the play the outlook is always “sunny” or “rainy”, they all belong to cluster 3.

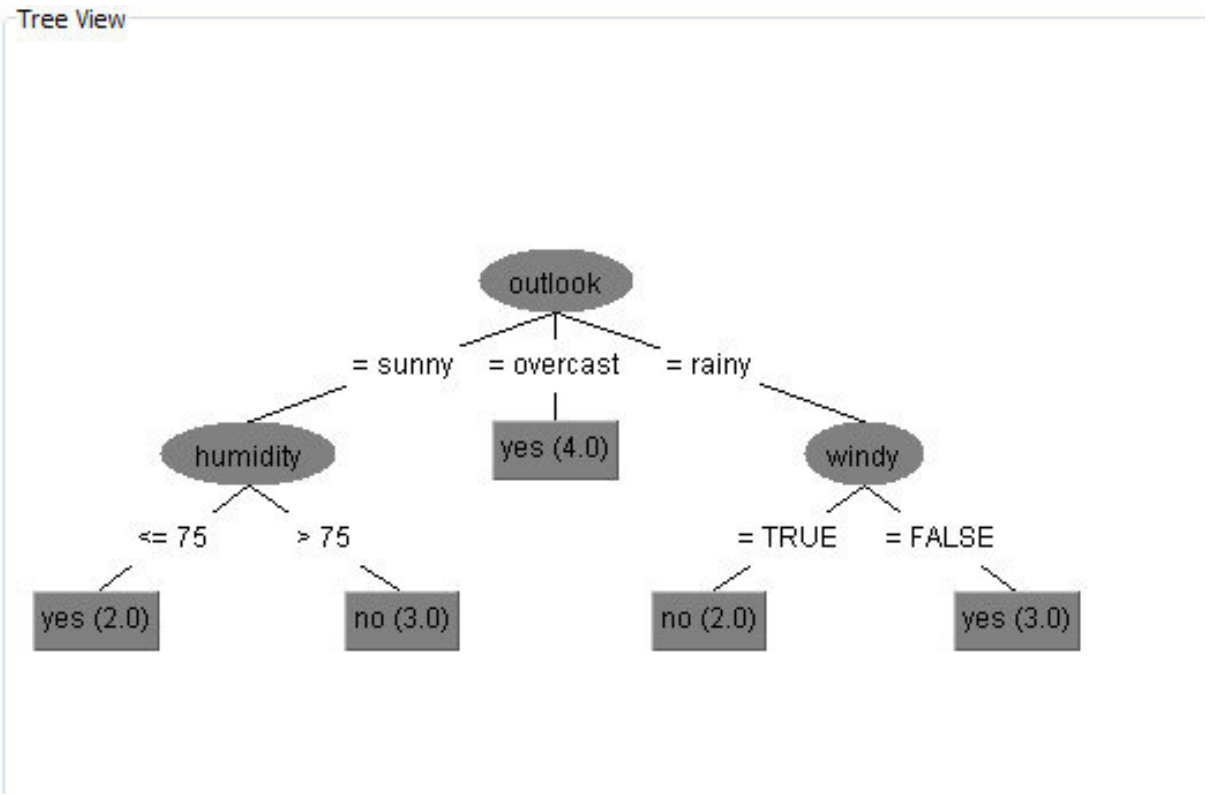
Another interesting graph for cluster I found was *windy vs. play*:




Here in this graph we can see the complete separation of clusters in the four corners. Very Interesting indeed.

Next I ran J48:

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                          -0.1395
Mean absolute error                       0.4643
Root mean squared error                   0.6116
Relative absolute error                  98.4466 %
Root relative squared error             125.5591 %
Total Number of Instances                14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.667     0.8        0.6       0.667     0.632      0.433    yes
                 0.2       0.333      0.25      0.2       0.222      0.433    no
Weighted Avg.    0.5       0.633      0.475     0.5       0.485      0.433

=== Confusion Matrix ===

 a b   <-- classified as
 6 3 | a = yes
 4 1 | b = no
```




However, this tree is not very accurate in predicting “no” and is not very accurate. Only 50% of the dataset is correctly predicted.

For the Rule based classifiers, I used the aarf file that have been discretized the numeric datasets. I will only show the attribute “Play” to predict “yes” or “no”. The classifiers I ran are Conjunctive Rule, JRip, and PART. Here is what the explanation of what they are.

**Conjunctive Rule**:
*This class implements a single conjunctive rule learner that can predict for numeric and nominal class labels. A rule consists of antecedents "AND"ed together and the consequent (class value) for the classification/regression. In this case, the consequent is the distribution of the available classes (or numeric value) in the dataset. If the test instance is not covered by this rule, then it's predicted using the default class distributions/value of the data not covered by the rule in the training data. This learner selects an antecedent by computing the Information Gain of each antecendent and prunes the generated rule using Reduced Error Prunning (REP). For classification, the Information of one antecedent is the weighted average of the entropies of both the data covered and not covered by the rule. For regression, the Information is the weighted average of the mean-squared errors of both the data covered and not covered by the rule. In pruning, weighted average of accuracy rate of the pruning data is used for classification while the weighted average of the mean-squared errors of the pruning data is used for regression.*

**JRip:**
*This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which is proposed by William W. Cohen as an optimized version of IREP.*

*The algorithm is briefly described as follows:*

*Initialize RS = {}, and for each class from the less prevalent one to the more frequent one, DO:*

*1. Building stage: repeat 1.1 and 1.2 until the descrition length (DL) of the ruleset and examples is 64 bits greater than the smallest DL met so far, or there are no positive examples, or the error rate >= 50%.*

*1.1. Grow phase:*
*Grow one rule by greedily adding antecedents (or conditions) to the rule until the rule is perfect (i.e. 100% accurate). The procedure tries every possible value of each attribute and selects the condition with highest information gain: p(log(p/t)-log(P/T)).*

*1.2. Prune phase:*
*Incrementally prune each rule and allow the pruning of any final sequences of the antecedents;*
*The pruning metric is (p-n)/(p+n) -- but it's actually 2p/(p+n) -1, so in this implementation we simply use p/(p+n) (actually (p+1)/(p+n+2), thus if p+n is 0, it's 0.5).*

*2. Optimization stage: after generating the initial ruleset {Ri}, generate and prune two variants of each rule Ri from randomized data using procedure 1.1 and 1.2. But one variant is generated from an empty rule while the other is generated by greedily adding antecedents to the original rule. Moreover, the pruning metric used here is (TP+TN)/(P+N).*
*Then the smallest possible DL for each variant and the original rule is computed. The variant with the minimal DL is selected as the final representative of Ri in the ruleset.*
*After all the rules in {Ri} have been examined and if there are still residual positives, more rules are generated based on the residual positives using Building Stage again.*

*3. Delete the rules from the ruleset that would increase the DL of the whole ruleset if it were in it. and add resultant ruleset to RS.*

*ENDDO*

**PART:**

Class for generating a PART decision list. For more information, see

Eibe Frank and Ian H. Witten (1998). Generating Accurate Rule Sets Without Global Optimization. In Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, Morgan Kaufmann Publishers, San Francisco, CA.

```
=== Run information ===

Scheme:       weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===


Single conjunctive rule learner:
--------------------------------
 => play = yes

Class distributions:
Covered by the rule:
yes     no
0.6     0.4

Not covered by the rule:
yes     no
0       0

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0
Mean absolute error                      0.4719
Root mean squared error                  0.4932
Relative absolute error                100.0693 %
Root relative squared error            101.2513 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.643      1         0.783      0.389    yes
                 0         0          0          0         0          0.389    no
Weighted Avg.    0.643     0.643      0.413      0.643     0.503      0.389

=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 5 0 | b = no
```

Here the single conjunctive rule learner assume to always assume “yes” for the attribute “Play”. In the case of this dataset, to always assume “yes” actually yield high accuracy for classifier predictions since 9 out of 14 elements fire the rule.

```
=== Run information ===

Scheme:       weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:
===========

(humidity = high) and (outlook = sunny) => play=no (3.0/0.0)
(outlook = rainy) and (windy = TRUE) => play=no (2.0/0.0)
 => play=yes (9.0/0.0)

Number of Rules : 3


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           6               42.8571 %
Incorrectly Classified Instances         8               57.1429 %
Kappa statistic                         -0.3659
Mean absolute error                      0.5915
Root mean squared error                  0.6624
Relative absolute error                125.4104 %
Root relative squared error            135.9865 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.667     1          0.545     0.667     0.6         0.267    yes
                 0         0.333      0         0         0           0.267    no
Weighted Avg.    0.429     0.762      0.351     0.429     0.386       0.267

=== Confusion Matrix ===

 a b   <-- classified as
 6 3 | a = yes
 5 0 | b = no
```

The JRip classifier has 42.85% for predicting correctly whether to play or not. There are 2 rules to fire for “no” otherwise “yes” because there are far more “yes” data points and “no”. I think because “no” rules takes precidence over “yes” rule, and non of the “no” rules has correctly predicted the outcome.

```
=== Run information ===

Scheme:       weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

PART decision list
------------------

outlook = overcast: yes (4.0)

humidity = high: no (5.0/1.0)

: yes (5.0/1.0)

Number of Rules  :       3


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           6               42.8571 %
Incorrectly Classified Instances         8               57.1429 %
Kappa statistic                         -0.1429
Mean absolute error                      0.5512
Root mean squared error                  0.6581
Relative absolute error                116.8738 %
Root relative squared error            135.1168 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.444     0.6         0.571     0.444     0.5         0.411    yes
                   0.4       0.556       0.286     0.4       0.333       0.411    no
Weighted Avg.      0.429     0.584       0.469     0.429     0.44        0.411

=== Confusion Matrix ===

 a b   <-- classified as
 4 5 | a = yes
 3 2 | b = no
```

Very much like the Tree from above, the “yes” predictions that’s correctly done because “Overcast” would always yield “yes” on “Play” in the dataset. As for other rules, they are not accurate in prediction. The total accuracy is only 42.85%

Below are expanded experiment on the same attribute “Play” with the nearest neighbor algorithms NNge, IBk, and Kstar.

```
Scheme:       weka.classifiers.rules.NNge -G 5 -I 5
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===


NNGE classifier

Rules generated :
        class no IF : outlook in {rainy} ^ temperature in {mild,cool} ^ humidity in {high,normal} ^ windy in {TRUE}  (2)
        class yes IF : outlook in {overcast,rainy} ^ temperature in {hot,mild,cool} ^ humidity in {high,normal} ^ windy in {FALSE}  (5)
        class yes IF : outlook in {overcast} ^ temperature in {mild,cool} ^ humidity in {high,normal} ^ windy in {TRUE}  (2)
        class yes IF : outlook in {sunny} ^ temperature in {mild,cool} ^ humidity in {normal} ^ windy in {TRUE,FALSE}  (2)
        class no IF : outlook in {sunny} ^ temperature in {hot,mild} ^ humidity in {high} ^ windy in {TRUE,FALSE}  (3)

Stat :
        class yes : 3 exemplar(s) including 3 Hyperrectangle(s) and 0 Single(s).
        class no : 2 exemplar(s) including 2 Hyperrectangle(s) and 0 Single(s).

        Total : 5 exemplars(s) including 5 Hyperrectangle(s) and 0 Single(s).

        Feature weights : [0.24674981977443894 0.029222565658954577 0.15183550136234153 0.04812703040826924]


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                          0.0667
Mean absolute error                      0.4286
Root mean squared error                  0.6547
Relative absolute error                 90.8738 %
Root relative squared error            134.4062 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.667    0.6        0.667     0.667     0.667      0.533    yes
                 0.4      0.333      0.4       0.4       0.4        0.533    no
Weighted Avg.    0.571    0.505      0.571     0.571     0.571      0.533

=== Confusion Matrix ===

 a b   <-- classified as
 6 3 | a = yes
 3 2 | b = no
```

This is the nearest neighbor classifier. I predicted correctly 57.14% of the time. It predicts better than most of the other rules from above but the predicates of the rule is not simplified enough. It doesn’t realize, for example,  that all “Overcast” in outlook result in “yes” in play.

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 2 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 2 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          10               71.4286 %
Incorrectly Classified Instances         4               28.5714 %
Kappa statistic                          0.3171
Mean absolute error                      0.471
Root mean squared error                  0.5192
Relative absolute error                 99.8767 %
Root relative squared error            106.5887 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.889     0.6        0.727     0.889     0.8        0.667    yes
                 0.4       0.111      0.667     0.4       0.5        0.667    no
Weighted Avg.    0.714     0.425      0.706     0.714     0.693      0.667

=== Confusion Matrix ===

 a b   <-- classified as
 8 1 | a = yes
 3 2 | b = no
```

This is the IB/k classifier with 2 nearest neighbor algorithm. It has 71.42% accuracy in prediction, better at predicting “yes” for the Play attribute than “no”. There are far more “yes” than “no” so the accuracy remains high.

```
=== Run information ===

Scheme:       weka.classifiers.lazy.KStar -B 20 -M a
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

KStar Beta Verion (0.1b).
Copyright (c) 1995-97 by Len Trigg (trigg@cs.waikato.ac.nz).
Java port to Weka by Abdelaziz Mahoui (am14@cs.waikato.ac.nz).

KStar options : -B 20 -M a

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                          0.0667
Mean absolute error                      0.5035
Root mean squared error                  0.5716
Relative absolute error                106.7716 %
Root relative squared error            117.362  %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.667     0.6         0.667     0.667     0.667      0.444    yes
                 0.4       0.333       0.4       0.4       0.4        0.444    no
Weighted Avg.    0.571     0.505       0.571     0.571     0.571      0.444

=== Confusion Matrix ===

 a b   <-- classified as
 6 3 | a = yes
 3 2 | b = no
```

This is the KStar result with the common result of 57.14% again with the similar results to other classification algorithms. Once again its better in predicting “yes” in Play rather than “no” in Play.